CSSE1001

Semester 2, 2014

Assignment 2

10 marks

Due Thursday 16 October, 2014, 9:30am

A GUI for plotting PV data

1 Introduction

In assignment 1 you wrote a simple tool for getting data from PV arrays. For this assignment you will write a GUI for plotting this data. A user of your program will be able to choose a date, which PV array to look at and what data is to be plotted.

2 Assignment Tasks

For each class and method that you write you need to provide a suitable comment giving a description and where necessary the type and any preconditions. You should use the triple-quote commenting style.

2.1 Download file

The file `assign2.py` is for your assignment. Add your name and student number in the space provided. When you have completed your assignment you will submit the file `assign2.py` containing your solution to the assignment.

The file already contains some code (in two parts). Do not modify any of this code! The first part should be left at the beginning of the file and consists of some import statements and support code. The second part should appear at the end of your code. **NOTE:** You will lose marks if you don't follow these instructions.

2.2 Write the code

Finally, write your solution to the assignment making sure you have included suitable comments. Your solution should include at least the following classes.

2.2.1 PVData Class

This class is used to hold the PV data for a given date. This class must define at least the following methods. You are required to use a dictionary to store the array power data. You should also use separate lists to store the temperature and sunlight data.

- `__init__(self)` that initializes the PV data to use yesterday's data.
- `change_date(self, date)` changes the data to be for the given date (making this date the 'current date'). If the same date is used (i.e. no change) then it **must not** extract the same data from the server again.
- `get_date(self)` that returns the date for the stored data.
- `get_time(self, time_index)` that returns the time for the given index of the time data.
- `get_temperature(self)` that returns the list of temperature values for the current date.
- `get_sunlight(self)` that returns the list of sunlight values for the current date.
- `get_power(self, array)` that returns the list of power output for the current date and the the given array (the array name).
- `get_cumulative_energy(self, array)` **(for CSSE7030 students only)** computes and returns the list of cumulative power values for the current date and the the given array (the array name).

Example (given the length of data, we have used slicing below to extract a small sublist, also only the error messages are displayed - not the traceback)

```
>>> pvd = PVData()
>>> pvd.change_date('13-09-2014')
>>> pvd.get_date()
'13-09-2014'
>>> pvd.get_time(300)
'10:00'
>>> pvd.get_temperature()[300:304]
[22.0, 22.0, 21.5, 21.5]
>>> pvd.get_sunlight()[300:304]
[252.7, 252.7, 216.3, 216.3]
>>> pvd.get_power('UQ Centre, St Lucia')[300:304]
[80495, 74765, 68285, 65250]
>>> pvd.get_power('All Arrays Combined')[300:304]
[288155, 270165, 248465, 239472]
>>> pvd.get_cumulative_energy('All Arrays Combined')[300:304]
[99963090, 100233255, 100481720, 100721192]
>>> get_data_for_date('')
ValueError: No date given.

>>> get_data_for_date('20-10-2050')
ValueError: Date must be before today.

>>> get_data_for_date('20_10_2014')
ValueError: Invalid date: '20_10_2014'

>>>
```

2.2.2 Plotter Class

This class is responsible for doing the plotting and should inherit from `Canvas`.

2.2.3 OptionsFrame Class

This class is the 'widget' used for choosing options and should inherit from `Frame`. It consists of the following:

For CSSE1001 students

- A row of three `Checkbutton`'s so the user can choose what data is to be displayed (Power, Temperature and Sunlight).
- An `Entry` box where the user can enter a date.
- A `Button` to apply the choice of date.
- An `OptionMenu` allowing the user to choose which array to display data for. (The default choice is All Arrays Combined.)

For CSSE7030 students

- A row of three `Radiobutton`'s to choose between No Power, Instantaneous Power and Cumulative Power.
- A row of two `Checkbutton`'s so the user can choose what data (other than power) is to be displayed (Temperature and Sunlight).
- An `Entry` box where the user can enter a date.
- A `Button` to apply the choice of date and array choice.
- An `OptionMenu` allowing the user to choose which array to display data for. (The default choice is All Arrays Combined.)

2.2.4 PVPlotApp Class

This is the top-level class for the GUI. It is responsible for creating and managing instances of the above classes.

At the top of the application the current date is displayed and if the left mouse button is clicked (and held down) a vertical line is drawn on the canvas and the data at that time, along with the time, is displayed next to the date. Dragging will shift the line and change the data displayed. The displayed data values should correspond to the values which are visible in the plot (e.g. only show the temperature at the top if the temperature checkbox is selected).

Note that, if the user enters an invalid date string, or if the date is valid but out of range (e.g. in the future) then the support file code will raise an exception. You need to catch such exceptions and display them in a po-up `tkMessageBox` window.

Details of the required GUI layout is shown in the examples listed below. CSSE1001 and CSSE7030 students must adhere to the CSSE1001 layout and CSSE7030 layout respectively.

NOTE: You must use pack (rather than grid) to do your GUI layout.

2.3 Examples

The course web page contains the following examples.

- Screenshots showing the GUI in use (CSSE1001 layout)
- Screenshots showing the GUI in use (CSSE7030 layout)
- Screencast showing the application in action.

2.4 Hints

For `OptionsFrame`, when the user changes the date or the PV array the system needs to possibly fetch new data and to then display the required data. In order to do this you need to call a method of `PVPlotApp`. This can be achieved in one of two ways - make the `PVPlotApp` object or the required method an argument to the `OptionsFrame` constructor.

For `Plotter` you will need to access the plot data. Probably the simplest approach is to make the `PVData` object an argument to the `Plotter` constructor. You might also find it useful to access the `OptionsFrame` object and a method to change what is displayed at the top of the application window.

For redrawing, either because the window has been resized or because the data to be displayed has changed, it's easiest to delete everything on the canvas (`delete(ALL)`) and draw all the required lines and polygons again.

3 Assessment and Marking Criteria

In addition to providing a working solution to the assignment problem, the assessment will involve discussing your code submission with a tutor. This discussion will take place in the practical session you have signed up to in week 12. You **must** attend that session in order to obtain marks for the assignment.

In preparation for your discussion with a tutor you may wish to consider:

- any parts of the assignment that you found particularly difficult, and how you overcame them to arrive at a solution;
- whether you considered any alternative ways of implementing a given function;
- where you have known errors in your code, their cause and possible solutions (if known).

It is also important that you can explain to the tutor how each of the functions that you have written operates (for example, if you have used a for loop or a while loop in a function, why this was the right choice).

Marks will be awarded based on a combination of the correctness of your code and on your understanding of the code that you have written. A technically correct solution will not elicit a pass mark unless you can demonstrate that you understand its operation.

We will mark your assignment according to the following criteria.

Criteria	**Mark**
Your code is mostly complete, correct, clear, succinct and well commented. You are able to explain your code.	8 - 10
Your code has some problems OR you have some problems explaining your code.	4 - 7
Your code is clearly incomplete, incorrect, too complex or hard to understand OR you have major problems explaining your code.	1 - 3
Your work has little or no academic merit.	0

A partial solution will be marked. If your partial solution causes problems in the Python interpreter please comment out that code and we will mark that.

Please read the section in the course profile about plagiarism.

4 Assignment Submission

You must submit your completed assignment electronically through Blackboard.

Please read
`http://www.library.uq.edu.au/ask-it/blackboard-assessment`
for information on submitting through Blackboard.

You should electronically submit your copy of the file `assign2.py` (use this name - all lower case).

You may submit your assignment multiple times before the deadline - only the last submission will be marked. After each submission please use Blackboard to check that the file you submitted was the one you intended to submit. Make sure the file is called `assign2.py` and not, for example, `assign2.py.py`

Late submission of the assignment will not be accepted. In the event of exceptional personal or medical circumstances that prevent you from handing in the assignment on-time, you should contact the lecturer in charge and be prepared to supply appropriate documentary evidence. You should be prepared to submit whatever work you have completed at the deadline, if required. Requests for extensions should be made as soon as possible, and preferably before the assignment due date.